Amendments to the CLAIMS

Please amend claims 7, 8, 9, 10, 27, 32, 33, 34, 35, 36 and 53 as follows.

1. (previously presented) In an electronic device, a method for communicating with one or more control instruments, the method comprising:

providing a common communication interface for communicating with a first control instrument via a driver for driving hardware of the first control instrument, wherein the driver is specific to the hardware of the first control instrument, wherein the common communication interface includes a command interpreter for generating a command for the first control instrument that is not recognized in the driver;

receiving a first creation command;

establishing a first communication channel linking the command interpreter and the first control instrument in response to the first creation command; and

enabling the command interpreter to communicate with the first control instrument independently of an interface bus standard type and an interface hardware driver type by converting the command for the first control instrument generated from the command interpreter to a command for the first control instrument that is recognized in the driver.

2. (original) The method of claim 1, wherein receiving the first creation command comes from a user interface.

3. (original) The method of claim 1 further comprising:

establishing a second communication channel linking the command interpreter and a second control instrument in response to a second creation command from the user interface.

4. (original) The method of claim 3, wherein the first communication channel is established through a first communication interface of the first control instrument and the second communication channel is established through a second communication interface of the second control instrument, the first communication interface being of a first type and the second communication interface being of a second type.

5. (original) The method of claim 4, wherein the first communication interface type and the second communication interface type include any of the supported interface types.

6. (original) The method of claim 5, wherein at least one of the first and second communication interface type is Virtual Instrumentation Software Architecture (VISA).

7. (currently amended) The method of claim 4, wherein:

the first control instrument having a communication interface is selected from a group of instrument interfaces having a first driver that includes the first type of communication interface; ~~and,~~ and

the second control instrument having a communication interface is selected from a group of instrument interfaces having a second driver that includes the second type of communication interface.

8. (currently amended) The method of claim 3 further comprising:

establishing the first communication channel with the first control instrument in response to a first instantiation command according to a first syntax; ~~and,~~ and

establishing the second communication channel with the second control instrument in response to a second instantiation command according to the first syntax.

9. (currently amended) The method of claim 3 further comprising:

creating a first instrument object associated with the first communication channel in response to an interpreter command, wherein the first instrument object has properties;

creating a second instrument object associated with the second communication channel in response to the interpreter command, wherein the second instrument object has properties;

creating an object array including the first instrument object and the second instrument object as elements of the object array in response to an array creation command to the

command interpreter, wherein the object array comprises properties; ~~and,~~ and

changing the properties of the first communication channel and the second communication channel in response to the interpreter command to change the properties of the object array.

10. (currently amended) The method of claim 9 further comprising:

changing the configuration of the first communication channel in response to the interpreter command to change the properties of the first instrument object; ~~and,~~ and

changing the configuration of the second communication channel in response to the interpreter command to change the properties of the second instrument object.

11. (original) The method of claim 9, wherein the first instrument object has a read function, the function further comprising:

receiving data from the first communication channel in response to the interpreter command to execute the read function of the first instrument object.

12. (original) The method of claim 9, wherein the first instrument object has a write function, the function further comprising:

transmitting data through the first communication channel in response to the interpreter command to execute the write function of the first instrument object.

13. (original) The method of claim 9 further comprising:

displaying the configuration of the first communication channel in response to the interpreter command to display the properties of the first instrument object.

14. (original) The method of claim 1 further comprising:

detecting an available interface for the first communication channel with the first control instrument, wherein the first communication channel is established on a detected interface.

15. (original) The method of claim 1, wherein the common communication interface includes a command interpreter having an instrument engine operating in an array-based environment.

16. (original) The method of claim 15 further comprising generating timer events and event handling operations.

17. (original) The method of claim 15 further comprising restoring an object to the array-based environment.

18. (original) The method of claim 15 further comprising buffering data between the interface hardware and the user interface.

19. (original) The method of claim 15 further comprising creating record files for data transfer.

20. (original) The method of claim 15 further comprising validating parameters.

21. (original) The method of claim 15 further comprising byte swapping.

22. (original) The method of claim 15 further comprising configuring object properties.

23. (original) The method of claim 15 further comprising translating error codes.

24. (original) The method of claim 15 further comprising data type casting.

25. (original) A method of claim 1, wherein the first communication channel is established by linking a compilation means and the first control instrument in response to the first creation command independent of an interface bus standard type and an interface hardware driver type.

26. (original) The method of claim 25, wherein the compilation means compiles a user created program to a stand-alone executable file when a command for compiling the program is received.

27. (currently amended) A system for communicating with one or more control instruments, the communication system comprising:

a user interface adapted to receive a first creation command;

a common communication interface for communicating with a first control instrument via a driver for driving hardware of the first control instrument, wherein the driver is specific to the hardware of the first control instrument, wherein the common communication interface includes,

a command interpreter adapted to receive the first creation command and generate a command for the first control instrument that is not recognized in the driver, and

an adaptor for converting the command for the first control instrument generated from the command interpreter to a command for the first control instrument that is recognized in the driver; and

a first communication channel linking the command interpreter and the first control instrument.

28. (previously presented) The system of claim 27, further comprising:

a second control instrument; and

a second communication channel linking the command interpreter and the second control instrument.

29. (previously presented) The system of claim 28, wherein the first communication channel is established through a first communication interface of the first control instrument and the second communication channel is established through a second communication interface of the second control instrument, the first communication interface being of a first type and the second communication interface being of a second type.

30. (previously presented) The system of claim 29, wherein the first communication interface type and the second communication interface type include any of the supported interface types.

31. (previously presented) The system of claim 30, wherein at least one of the first and second communication protocols is Virtual Instrumentation Software Architecture (VISA).

32. (currently amended) The system of claim 28, wherein:

the first control instrument having a communication interface is selected from a group of instrument interfaces supplied by a first driver that includes the first type of communication interface; ~~and,~~ and

the second control instrument having a communication interface is selected from a group of instrument interfaces supplied by a second driver that includes the second type of communication interface.

33. (currently amended) The system of claim 32, wherein:

the first communication channel is established with the first control instrument in response to a first instantiation command according to a first syntax; ~~and,~~ and

the second communication channel is established with the second control instrument in response to a second instantiation command according to the first syntax.

34. (currently amended) The system of claim 28, further comprising:

a first configuration command according to a second syntax, for changing a configuration of the first communication channel with the first control instrument in response to the first configuration command; ~~and,~~ and

a second configuration command according to a second syntax, for changing a configuration of the second communication channel with the second control instrument in response to the second configuration command.

35. (currently amended) The system of claim 28, further comprising:

a first instrument object associated with the first communication channel in response to an instrument command;

a second instrument object associated with the second communication channel in response to the instrument command;

an object array including the first instrument object and the second instrument object as elements of the object array in response to an array creation command to the command interpreter; ~~and,~~ and

means for changing the properties of the first communication channel and the second communication channel in response to the interpreter command to change a property of the object array.

36. (currently amended) The system of claim 28, further comprising:

a first instrument object associated with the first communication channel, wherein the first instrument object has properties; ~~and,~~ and

means for changing the configuration of the first communication channel in response to an interpreter command to change the properties of the first instrument object.

37. (previously presented) The system of claim 36, wherein the first instrument object has a read function to receive data from the first communication channel in response to the interpreter command to execute the read function of the first instrument object.

38. (previously presented) The system of claim 36, wherein the first instrument object has a write function to transmit data through the first communication channel in response to the interpreter command to execute the write function of the first instrument object.

39. (previously presented) The system of claim 37, further comprising:

means for displaying the configuration of the first communication channel in response to the interpreter command to display the properties of the first instrument object.

40. (previously presented) The system of claim 27, further comprising:

means for detecting an available interface for the first communication channel with the first control instrument, wherein the first communication channel is established on a detected interface.

41. (original) The system of claim 27 further comprising an instrument engine operating in an array-based environment.

42. (original) The system of claim 41, wherein the instrument engine is adapted to generate timer events and event handling operations.

43. (original) The system of claim 41, wherein the instrument engine is adapted to restore an object to the array-based environment.

44. (original) The system of claim 41, wherein the instrument engine is adapted to buffer data between the interface hardware and the user interface.

45. (original) The system of claim 41, wherein the instrument engine is adapted to create record files for data transfer.

46. (original) The system of claim 41, wherein the instrument engine is adapted to validate parameters.

47. (original) The system of claim 41, wherein the instrument engine is adapted to perform byte swapping operations.

48. (original) The system of claim 41, wherein the instrument engine adapted to configure object properties.

49. (original) The system of claim 41, wherein the instrument engine is adapted to translate error codes.

50. (original) The system of claim 41, wherein the instrument engine is adapted to perform data type casting operations.

51. (previously presented) The system of claim 27, wherein the first communication channel is established by linking a compilation means and the first control instrument in response to the first creation command independent of an interface bus standard type and an interface hardware driver type.

52. (original) The system of claim 51, wherein the compilation means compiles a user created program to a stand-alone executable file when a command for compiling the program is received.

53. (currently amended) In an electronic device, a method for communicating a control instrument, the method comprising:

instantiating an instrument object in response to an instantiating function call;

establishing a communication channel linking a control instrument to the instrument object in response to a function call for establishing the communication channel;

writing and reading data between the control instrument and the instrument object in response to write and read function calls, wherein the write and read function calls are converted

into write and read function calls that are specific to the control instrument using the instrument object; ~~and,~~ and

disconnecting the instrument object from the control instrument in response to a close function call.

54. (previously presented) A system for communicating with a control instrument, comprising:

means for receiving a creation command from a user interface;

means for establishing a communication channel linking a command interpreter and the control instrument in response to the creation command, wherein the command interpreter generates a first command for communicating with the control instrument that is not specific to the control instrument;

means for creating an object array including a first instrument object and a second instrument object as elements of the object array in response to an array creation command to the command interpreter, wherein the object array comprises properties; and

means for converting the first command into a second command for communicating with the control instrument using the object array, wherein the second command is specific to the control instrument.

55. (previously presented) A computer program product, tangibly stored in a computer readable medium comprising instructions operable by a command interpreter in response to commands to the interpreter, the instructions causing the command interpreter to:

providing a common communication interface for communicating with a first control instrument via a driver for driving hardware of the first control instrument, wherein the driver is specific to the hardware of the first control instrument, wherein the common communication interface includes a command interpreter for generating a command for the first control instrument that is not recognized in the driver;

receive a first creation command from a user interface;

establish a first communication channel linking the command interpreter and a first control instrument in response to the first creation command; and

enabling the command interpreter to communicate with the first control independently of an interface bus standard type and an interface hardware driver type by converting the command

for the first control instrument generated from the command interpreter to a command for the first control instrument that is recognized in the driver.